# A Copy-Augmented Generative Model for Open-Domain Question Answering

**Anonymous ACL submission**

## Abstract

Open-domain question answering is a challenging task with a wide variety of practical applications. Existing modern approaches mostly follow a standard two-stage paradigm: retriever then reader. In this article, we focus on improving the effectiveness of the reader module and propose a novel copy-augmented generative approach that integrates the merits of both extractive and generative readers. In particular, our model is built upon the powerful generative model FiD (Izacard and Grave, 2020b). We enhance the original generative reader by incorporating a pointer network to encourage the model to directly copy words from the retrieved passages. We conduct experiments on the two benchmark datasets, Natural Questions and TriviaQA, and the empirical results demonstrate the performance gains of our proposed approach.

| |
|---|
| **Question:** where was a hologram for the king filmed?<br>**Passages (Truncated):** title: A Hologram for the King (film) context: Production was set to begin in first quarter of 2014. Principal photography commenced on March 6, 2014 in Morocco. Filming also took place in Hurghada in Egypt, as well as in Berlin and Düsseldorf in Germany. Shooting wrapped in June 2014.<br>**Answer:** Hurghada in Egypt, Berlin and Düsseldorf in Germany |
| **FiD:** Dubai in Germany<br>**FiD-PGN:** Hurghada in Egypt |
| **Question:** who has the most trophies in la liga?<br>**Passages (Truncated):** title: La Liga context: A total of 62 teams have competed in La Liga since its inception. Nine teams have been crowned champions, with Real Madrid winning the title a record 33 times and Barcelona 25 times.<br>**Answer:** Real Madrid |
| **FiD:** 33<br>**FiD-PGN:** Real Madrid |

Table 1: Comparisons of answers generated by FiD and our approach. The orange text represents supportive sentences.

## 1 Introduction

Open-domain question answering (ODQA) focuses on providing highly precise answers to natural language questions from a large collection of unstructured text data (Voorhees, 1999). With the pioneering work of DrQA (Chen et al., 2017), modern approaches to ODQA commonly adopt a simple two-stage *retriever-reader* pipeline, that firstly retrieve a relatively small number of support passages (Karpukhin et al., 2020; Yamada et al., 2021; Min et al., 2021b), followed by the reader identifying the answer.

The reader models can be broadly categorized into two classes: extractive (Chen et al., 2017; Asai et al., 2019; Karpukhin et al., 2020) and generative (Lewis et al., 2020a; Izacard and Grave, 2020b; Wu et al., 2021). Recently, benefiting from the powerful ability of large-scale pre-trained encoder-decoder language models (Raffel et al., 2019; Lewis et al., 2019) and the capability of aggregating information from multiple passages (Izacard and Grave, 2020b), generative approaches have achieved in general better performance than extractive methods.

Compared to extractive models, generative models generate text more freely, which makes it often suffer from the problem of producing hallucinated text that is inconsistent to the input or factual inaccuracy. This problem has been addressed in tasks like text summarization and machine translation (Maynez et al., 2020; Zhou et al., 2021). We found that the phenomenon also happens in ODQA. As shown in Table 1, the answer "Dubai in Germany" produced by the generative model FiD (Izacard and Grave, 2020b) is factual incorrect and the answer "33" in the second example is not coherent to the question. While in both cases, the ground-truth answers are present in the retrieved passages. Thus, we hypothesize that if we could put a constraint on the produced words to the input text, the generated answer will be more faithful.

Inspired by the work of See et al. (2017), we enhance the generative model with a pointer network (Vinyals et al., 2017), that enables the model

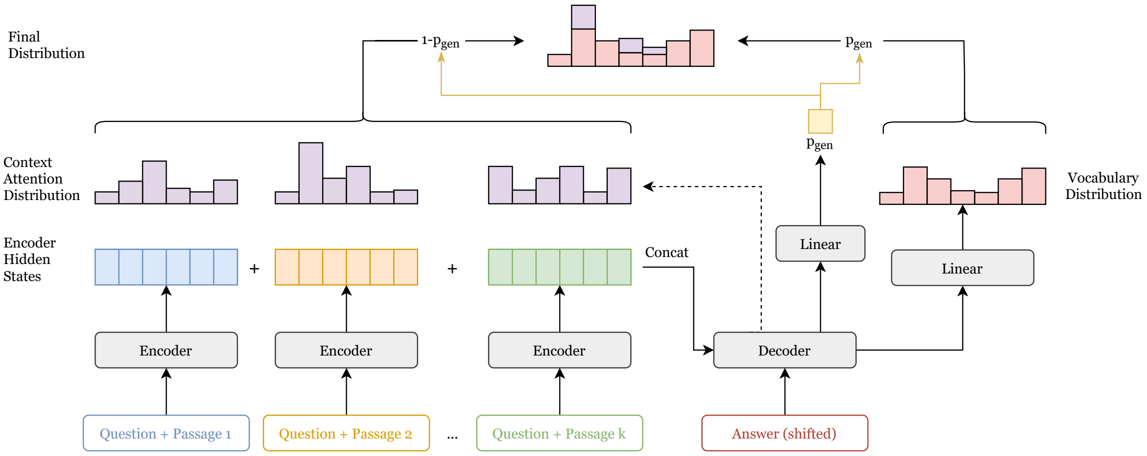

Figure 1: The overall architecture of our proposed model. We add a linear layer to calculate the generation probability, which decides the weights of generating words from vocabulary or copying from source passages.

to directly copy text from the retrieved passages while retains the ability of generating new words when the true answers are not explicitly present in the input. To be more specific, our model fusion-in-decoder pointer-generator network (FiD-PGN) is built upon the state-of-the-art model FiD. We reuse the encoder-decoder attention scores as the copy distribution to reduce the computational cost. Compared to FiD, we achieve comparative or even better accuracy on the Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) benchmarks, with less passages used in training. Our experiments results show the effectiveness and efficiency of our model.

## 2 Related Work

### 2.1 Open-Domain Question Answering

In this era of data explosion, ODQA offers a way to rapidly and accurately fulfill user's information needs, and hence has recently received significant attention from both industry and academia (Min et al., 2021a). Following the work of DrQA (Chen et al., 2017), most recent works build a two-stage *retriever-reader* system to tackle the problem. The retriever aims at retrieving supportive passages to the given question from a large document corpus. The reader intends to find answer of the question from the first stage retrieved passages. Early work of Chen et al. (2017) adapts a BiLSTM architecture with various lexical and semantic features from the question and passages as inputs. Later, with the emergence of large-scale pre-trained language models, readers based on pre-trained models such as BERT and T5 (Devlin et al., 2019; Raffel et al., 2019) have become a common approach (Yang et al., 2019; Karpukhin et al., 2020; Izacard and Grave, 2020b).

### 2.2 Generative Readers

Compared to extractive models which extract existing words from the retrieved passages, generative models are able to produce new words out of the retrieved passages, and thus provide a more flexible modeling framework. Min et al. (2020) and Lewis et al. (2020a) concatenate the given question with top retrieved passages and feed the concatenation to the BART model (Lewis et al., 2019). Izacard and Grave (2020b) separately encodes the question with each top retrieved passage, then takes the concatenation of the encoder outputs as input to the decoder. Their method provide a way to better aggregate evidence from multiple passages and improve the performance significantly. FiD-KD (Izacard and Grave, 2020a) is an extension of FiD model that increases the accuracy of passage retrieval by training the dense retriever with the guidance of the FiD reader iteratively.

### 2.3 Pointer-Generator Network

Pointer-Generator Network (See et al., 2017) is an extension of the sequence-to-sequence model by integrating a copy mechanism (Vinyals et al., 2017) into the generator. At each decoding stage, the model is able to either directly copy a word from the input or generate one with certain probability, and thus can be viewed as a combination of extractive and generative approaches. It has been frequently used in natural language tasks like summarization (Gu et al., 2016; See et al., 2017;

| Model | Reader Size | Top-$k$ | NQ | TriviaQA |
|---|---|---|---|---|
| DPR (BERT-base) (Karpukhin et al., 2020) | 110M | 24 | 41.5 | 57.9 |
| RAG-Seq (BART-large) (Lewis et al., 2020a) | 406M | 50 | 44.5 | 56.8 |
| FiD (T5-base) (Izacard and Grave, 2020b) | 220M | 100 | 48.2 | 65.0 |
| FiD-KD (T5-base) (Izacard and Grave, 2020a) | 220M | 100 | 49.6 | **68.8** |
| FiD-KD (Our implementation) | 220M | 25 | 48.5 | 67.5 |
| FiD-PGN | 220M | 25 | **51.4** | 68.4 |

Table 2: Exact match (EM) scores on NQ and TriviaQA test sets. Top-$k$ indicates the number of retrieved passages used during reader training. The performance of SOTA model is in **bold** and the second best model is in underline.

Gehrmann et al., 2018) and neural machine translation (Luong et al., 2014; Gu et al., 2017), but its application to ODQA has been less explored.

## 3 Method

Our model follows the standard two-stage *retriever-reader* framework with a focus on the enhancement of the reader module built upon the FiD model. We adopt the retriever results of FiD-KD, where a dense retriever similar to DPR (Karpukhin et al., 2020) is used. A pointer network is integrated into the FiD reader to facilitate copying words from the retrieved passages. The overall reader architecture is depicted in Figure 1.

**Reader Encoder.** The reader encoder of our model is identical to the one of FiD reader. We firstly concatenate the given question $q$ with each retrieved passage $p_i$ as $x_i = [q; p_i]$. Next, we pass each $x_i$ individually to the reader encoder, i.e., the encoder of T5 or BART model, and obtain the hidden representations $h_i = h_{i,1}, h_{i,2}, \ldots, h_{i,n}$ of the question-passage pair where $h_{i,j} \in \mathbb{R}^d$ and $d$ is the model dimension. Finally, we concatenate all the hidden representations $\{h_1, \ldots, h_k\}$ as input to the decoder.

**Reader Decoder.** Our approach mainly differs from FiD reader in the decoder module by adding a pointer network. Specifically, at each decoding step $t$, let $e_t \in \mathbb{R}^d$ be the embedding vector of the input token at this step, and denote $s_t^L \in \mathbb{R}^d$ as the output representation of the last layer $L$ of transformer decoder, then the probability of generation is given as follows,

$$p_{\text{gen}} = \sigma(w_e^T e_t + w_s^T s_t^L + b) \qquad (1)$$

where $w_e \in \mathbb{R}^d$, $w_s \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are all learnable parameters and $\sigma(\cdot)$ represents the sigmoid function. In addition, the probability of copying is $1 - p_{\text{gen}}$.

Next, let $\mathcal{V}$ denote the vocabulary containing words for the generative model and $|\mathcal{V}|$ be the size of the vocabulary. Then at step $t$, the probability distribution of words generation over the vocabulary is computed as,

$$P_{\text{vocab}} = \text{softmax}(W_E s_t^L) \qquad (2)$$

where $W_E \in \mathbb{R}^{|\mathcal{V}| \times d}$ is a learnable weight matrix.

Benefiting from the encoder-decoder attention layer in transformer architecture, we directly utilize the cross-attention score $\alpha_t^L$ of the last decoder layer $L$ over the source tokens for the target token $y_t$ as copy distribution. Then the probability of selecting $y_t$ in source sequence is calculated as,

$$P_{\text{ctx}}(y_t) = \sum\nolimits_{j: x_{1:k,j} = y_t} \alpha_{t,j}^L \qquad (3)$$

where $x_{1:k}$ denotes the concatenation of the top-$k$ retrieved passages, $x_{1:k,j}$ is the $j$-th token of $x_{1:k}$, and $\alpha_{t,j}^L$ is the $j$-th element of $\alpha_t^L$. If $y_t$ is not present in the top-$k$ retrieved passages, the $P_{\text{ctx}}(y_t)$ will be zero.

Finally, put all the above together, the target token $y_t$ could both be generated from vocabulary with probability $p_{gen}$, and copy from the source passages. The final prediction probability is defined as

$$P(y_t) = p_{\text{gen}} P_{\text{vocab}}(y_t) + (1 - p_{\text{gen}}) P_{\text{ctx}}(y_t). \quad (4)$$

## 4 Experiments

### 4.1 Datasets

We evaluate the performance of our approach on two standard ODQA datasets, NQ and TriviaQA. The NQ dataset comprises real queries that user issued on Google search engine along with answers. The TriviaQA dataset consists of question-answer pairs collected from trivia and quiz-league websites. The details of data statistics are listed at Appendix A. We use the data released on the repository of

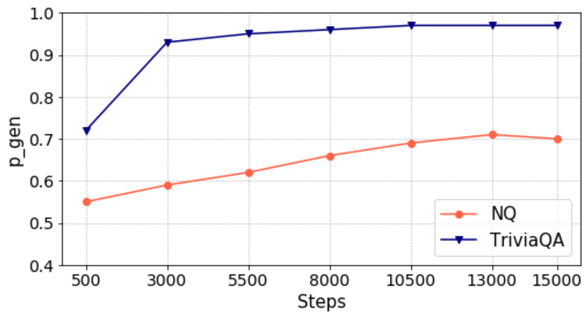

Figure 2: Generation probability $p_{\text{gen}}$ over training steps on NQ and TriviaQA.

FiD[1], containing question-answer pairs and top-100 passages retrieved by FiD-KD.

### 4.2 Implementation Details

We follow the experimental settings as in FiD. Our model is initialized with a pre-trained T5-base model, and trained using AdamW (Loshchilov and Hutter, 2017) algorithm with a learning rate of $10^{-4}$, linear scheduling with 15k total steps and 1k warm-up steps. Moreover, we train our model using the top-25 retrieved passages for each question and set the batch size as 64 due to computational limitation. All experiments are run on eight Nvidia V100 32GB GPUs.

### 4.3 Results

Table 2 shows the experimental results of our model and other approaches on the test sets, evaluated with the standard exact match (EM) score (Rajpurkar et al., 2016). For a fair comparison, we retrained the FiD reader on the top-25 retrieved passages to match our experimental settings. We show the results of different number of passages in Appendix B.

As shown in Table 2, our model outperforms FiD-KD on both NQ and TriviaQA datasets under the same setting. This demonstrates that the pointer network could help to generate answers more accurately. It is worth noting that, compared with FiD-KD trained with the top-100 retrieved passages, our model achieves comparative or even better results with only 1/4 of the input data and without introducing many parameters (only 1537 extra parameters are added), indicating the efficiency of our model.

## 5 Analysis

**Generation Probability.** We explore the probability of generation during training to further investigate the effects of the pointer module. As shown in Figure 2, the generation probability $p_{\text{gen}}$ in TriviaQA is always higher than the one in NQ. Note that a higher generation probability means that more tokens are produced from the vocabulary instead of copying from the input. We conjecture that this phenomenon is caused by the different question types. As stated in Rogers et al. (2021), Trivia questions are more like probing questions. Compared to the information-seeking questions in NQ, probing questions tend to need more complex reasoning, and thus it is difficult to directly extract relevant tokens from input texts. Moreover, this observation is also consistent with the results that the improvements of our model over FiD reader is smaller in TriviaQA than the one in NQ (0.9 vs. 2.9 EM for TriviaQA and NQ, respectively).

| Dataset | Overlap Type | FiD | FiD-PGN | Δ |
|---|---|---|---|---|
| NQ | Total | 48.5 | **50.6** | 2.1 |
| | Question Overlap | **73.5** | 70.1 | -3.4 |
| | Answer Overlap Only | 41.0 | **44.4** | 3.4 |
| | No Overlap | 28.8 | **32.4** | 3.6 |
| TriviaQA | Total | 67.5 | **68.4** | 0.9 |
| | Question Overlap | 88.4 | **89.6** | 1.2 |
| | Answer Overlap Only | 66.9 | **68.4** | 1.5 |
| | No Overlap | 41.5 | **43.4** | 1.9 |

Table 3: Test-train overlap evaluation on NQ and TriviaQA test sets.

**Test-Train Overlap Evaluation.** The study of test-train overlap (Lewis et al., 2020b) provides valuable insights into the model's question answering behavior. We evaluate our model on the same test data splits as in Lewis et al. (2020b). Table 3 reports the results with respect to three kinds of test-train overlaps. It can be seen that our approach improves most over FiD reader on "No Overlap" category, the most challenging setting, indicating a better generalization ability to question answering.

## 6 Conclusion

In this article, we propose a novel FiD-PGN approach for the reader module of ODQA under the standard *retriever-reader* framework. Specifically, we integrate a pointer network into the FiD reader to allow the model to directly select words from the retrieved passages. Experimental results show that our model outperforms FiD-KD on two benchmark datasets under the same setting, demonstrating the advantages of our method.

[1] https://github.com/facebookresearch/FiD

## A Statistics of datasets

The summary statistics of both datasets are shown in Table 4. It can be seen that TriviaQA has on average longer question length than NQ, indicating that questions in TriviaQA are relatively more complex.

| Statistics | NQ | TriviaQA |
|---|---|---|
| Train | 79,168 | 78,785 |
| Validation | 8,757 | 8,837 |
| Test | 3,610 | 11,313 |
| Avg. Qlen | 9.3 | 16.9 |
| Avg. Alen | 2.4 | 2.2 |

Table 4: Summary statistics of the two datasets. Avg. Qlen and Avg. Alen denote the average number of tokens per question and answer, respectively.

## B Training with Varying Number of Passages

Figure 3 shows the performance of our model and FiD reader with regard to different number of retrieved training passages. We train both models with top-$k$ passages ($k \in \{1, 5, 10, 25\}$) and evaluate on the development sets with the same number of passages. We can observe that the matching scores of both models increase with respect to the number of passages used in training, consistent with the findings in Izacard and Grave (2020b) that sequence-to-sequence model is capable of gathering information across multiple retrieved passages. Moreover, the two models show comparative performance when the number of training passages is small, but when more passages included, our model outperforms FiD, especially on the NQ dataset.

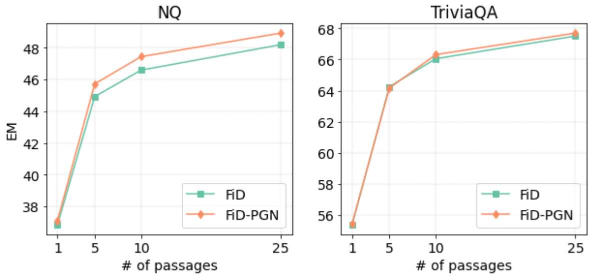

Figure 3: The variation of performance with different number of retrieved passages used in reader training. Exact match (EM) scores are measured on the development sets of NQ and TriviaQA.

## References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2019. Learning to retrieve reasoning paths over wikipedia graph for question answering. *CoRR*, abs/1911.10470.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *CoRR*, abs/1704.00051.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-up abstractive summarization. *CoRR*, abs/1808.10792.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. *CoRR*, abs/1711.02281.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *CoRR*, abs/1603.06393.

Gautier Izacard and Edouard Grave. 2020a. Distilling knowledge from reader to retriever for question answering. *CoRR*, abs/2012.04584.

Gautier Izacard and Edouard Grave. 2020b. Leveraging passage retrieval with generative models for open domain question answering. *CoRR*, abs/2007.01282.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *CoRR*, abs/1705.03551.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7(0):452–466.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019.

BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020a. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401.

Patrick S. H. Lewis, Pontus Stenetorp, and Sebastian Riedel. 2020b. Question and answer test-train overlap in open-domain question answering datasets. *CoRR*, abs/2008.02637.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *CoRR*, abs/1410.8206.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. 2020. On faithfulness and factuality in abstractive summarization. *CoRR*, abs/2005.00661.

Sewon Min, Jordan L. Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, Patrick S. H. Lewis, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Sejr Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen-tau Yih. 2021a. Neurips 2020 efficientqa competition: Systems, analyses and lessons learned. *CoRR*, abs/2101.00133.

Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. 2021b. Joint passage ranking for diverse multi-answer retrieval. *CoRR*, abs/2104.08445.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. *CoRR*, abs/2004.10645.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2021. QA dataset explosion: A taxonomy of NLP resources for question answering and reading comprehension. *CoRR*, abs/2107.12708.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2017. Pointer networks.

Ellen M. Voorhees. 1999. The TREC-8 question answering track report. In *Proceedings of The Eighth Text REtrieval Conference, TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999*, volume 500-246 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Yuxiang Wu, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2021. Training adaptive computation for open-domain question answering with computational constraints. *CoRR*, abs/2107.02102.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021. Efficient passage retrieval with hashing for open-domain question answering. *CoRR*, abs/2106.00882.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. *CoRR*, abs/1902.01718.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.